The Myth of the Best Practices Silver Bullet

Michael W. Evans and Corinne Segura
American Systems Corporation, Inc.

Frank Doherty
Space and Naval Warfare Systems Command

For many years, there has been much work attempting to identify a set of best practices that software-intensive projects could apply to aid in the acquisition, production, or upgrade of software. Spurred on by the 1987 and 2001 Defense Science Board findings, efforts conducted by the Software Engineering Institute and the Software Program Managers Network have identified and documented specific practices that have had apparent success in lowering project risk, improving cost and schedule performance, and enhancing user satisfaction. Since Section 804 of the National Defense Authorization Act for Fiscal Year 2003 was enacted on Dec. 2, 2002 and became law, there has been much activity in this area, particularly in the Department of Defense and its various services. This article explores some of these efforts, looks at the practices that have resulted, and attempts to examine certain key relationships that must be considered when applying them to projects.

Best practices are often looked on as the *Holy Grail* of process improvement, the silver bullet that will cure all ills. A manager might reasonably ask, "After all, couldn't I expect the same degree of success if I use the same processes in the management, engineering, assurance, and tracking of the project?" The answer is an unqualified *maybe*.

Indeed, far from being a silver bullet, there is some evidence that the term *best practices* lacks significant meaning. In 2001, Dr. Richard Turner conducted a study for the Department of Defense (DoD) [1] to identify credible best practices that could improve performance, predictability, quality, and operational effectiveness while lowering risk, shortening schedules, and reducing development costs. As a result of this study, Turner concluded that because the term *best practices* is consistently misused, it is misleading at best and useless at worst. It has become a catchall phrase that bundles diverse ideas about practices and frameworks, and is used by some to legitimize unproven practices, tools, or processes.

Unproven practices, while not designated as best, are often essential components of successful projects. They simply do not have a pedigree outside of the project to which they are being applied.

Practice Relationships

Software managers must not assume that just because a certain practice has been labeled *best* that it will indeed improve the performance, predictability, quality, and operational effectiveness of the software they are responsible for producing. Nor does it mean they should view these practices with outright suspicion, but only that they must understand the advantages and disadvantages of the practices for their particular projects and how they can be usefully adapted to the various needs of their organization.

To understand why so-called best practices are not a silver bullet, it is important to understand the difference between a process and a practice. A process is a set of interrelated resources and activities that transform inputs into outputs. When used in a consistently formal manner, these resources and activities tend to increase quality, shorten schedules, and lower cost and risk. Processes are used to conduct business, and they support a unique organizational culture. Practices are disciplines, methods, tools, or techniques that are used to accomplish a specific function or set of functions in a project environment. A process can include multiple practices.

A process can be considered a plan in that it describes what must be done to obtain an output and provides the framework needed to accomplish the necessary tasks. Practices define the manner in which the tasks must be conducted. Both are critical to a project's success because what is to be done must be planned, and how the plan will be accomplished must be defined. Moreover, practices must be adapted to the organization that uses them, and the relationships between the practices must be understood and managed if the expected benefits are to be realized. It is incumbent upon the manager to choose practices that are appropriate to the level of the organization that will implement them and adapt them as necessary. For example, a practice intended to meet the configuration management needs of the acquisition layer of an organization would not necessarily meet the needs of the development layer, but with a proper understanding of its uses, a manager could adapt the practice to meet the needs of both.

A typical program has several layers, each of which has different requirements and constraints regarding the processes and practices used and their implementation. The top layer, the user organization, requires deployment of a product or service that is responsive to the operational and support requirements of the user community. The needs of the user community form the baseline from which required practices are defined and implemented. The user organization requires practices that (1) capture, characterize, and control the user's operational requirements and constraints; (2) define interoperability and system interface requirements; (3) identify how these requirements are qualified and inserted into the operational environment; and (4) define how these elements are documented, maintained, and updated.

The next layer of the program, the acquisition organization, is chartered to acquire the right product at the lowest cost and in the shortest time to satisfy a specified user requirement. To do so, the acquisition organization works with the user to define what is required. It then converts the user needs into functional, engineering, product assurance and support requirements and constraints, and plans and implements processes – supported by practices – to acquire a system, software product, or services that will satisfy the user needs. The acquisition organization next prepares and issues documentation that establishes agreements between the acquirer and supplier(s), selects a supplier(s), and manages the acquisition process until the product or service is accepted.

The acquisition organization requires practices that facilitate the acquisition of the product or service at the lowest risk. It needs practices to collect and evaluate quantitative indicators of both project performance and product quality; to

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
SEP 2005		**00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The Myth of the Best Practices Silver Bullet	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Space and Naval Warfare Command San Diego,4301 Pacific Hwy,San Diego,CA,92110-3127**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
CROSSTALK The Journal of Defense Software Engineering, September 2005

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**5**	

assure the product's quality based on the quantitative indicators; to monitor developer performance; and to facilitate delivery, acceptance, and deployment of the product or service.

The next layer of the program, the development organization, is chartered to build a product in a manner that is consistent with established agreements and specifications and that maximizes profit and meets all commitments and agreements. The development organization needs practices that (1) specify architectures; (2) define, expand, and control specific engineering requirements that are traceable to those provided by the acquirer; (3) control and manage the development process; (4) monitor the quality of the products being developed or received from suppliers; (5) collect quantitative information from implemented practices to ascertain process effectiveness; (6) monitor cost and schedule performance; and (7) monitor development risk against progress toward established requirements.

Finally, support organizations – such as independent test, logistics, installation, product maintenance, etc. – require practices that enable processes, and provide management visibility into and control over the quality of the services provided and the risks that must be addressed.

Given these various needs, it is easy to see that a single set of best practices would be impossible to implement, and that any set of best practices must be adapted to the organizational layer that implements them. For example, best practices related to configuration management necessarily must be adapted to meet the needs of the user level, the supplier level, and the development level. Nevertheless, configuration management can be considered a necessary best practice that all organizations should address if they expect to succeed.

Once adapted to the needs of the various organizations, the practices must be integrated if the project is to progress effectively both within an organization and between organizations. For example, the requirements management process area may require effective implementation of the requirements definition, configuration management, defect identification and removal, and user involvement practice areas to accomplish the process.

In addition, the many practices a particular organization uses must interact with other practices, which are often unrelated, to provide seamless and effective support to the overall process. For example, if a particular practice identifies defects early in the development process, configuration management should account for this, and the software manager must adapt practices to remove defects accordingly to ensure potential efficiencies are realized.

Sample Practices

Table 1 on page 16 lists several project approaches or best practices that have originated from initiatives conducted by various organizations. This list is a small sample of the hundreds identified in the literature as being best [2]. These practices can be categorized by their typical application or use:

- **Policy Requirement.** A policy requirement defines a basic requirement that all program organizations must meet. This category focuses on a desired outcome and typically does not define specific processes or practices.
- **Organizational Concept.** These are general principles that are used to organize the project, allocate resources and responsibility, enable communications, and effect work assignment.
- **General Strategy.** This is a strategy that applies to all organizational components of a project but must be adapted to the needs of a particular organization to be effective. A general strategy, for example, might be that all projects apply continuous risk management to prevent negative consequences from unanticipated issues. Although related to the risk management practices of other organizations, specific implementation of the general strategy within an organization will depend on the organization's particular charter, culture, commitments, and constraints.
- **Business Strategy.** This is a strategy that defines how to accomplish specific business tasks.
- **Acquisition Framework.** This is a structure the acquisition organization uses to acquire, manage, and control the products or services and ensure they are responsive to user needs. It usually consists of activities, specifications, reviews, and reporting requirements.
- **Acquisition Strategy.** An overall statement of how an organization will acquire products and services consistent with user needs and requirements is an acquisition strategy. Expressed in general terms, it typically describes requirements that will constrain the selection and adaptation of processes and practices and defines the goals that must be met to satisfy validated needs as well as to maximize affordability.
- **General Practice.** This is a practice that supports every organizational level. Specific application of the practice will differ according to the needs of the user, acquirer, supplier, and support layers of the program organization.
- **Development Practice.** This is a practice that predominately supports the supplier's requirements.
- **Acquisition Practice.** This is a practice that ensures an acquisition is conducted effectively by the acquisition organization as it monitors the project and controls the suppliers. The practices are structured to evaluate and receive products and services rather than develop and deliver them.
- **Maturity Model.** This is used to evaluate the process maturity of an organization to determine the potential risk of a process and the potential to use it successfully in other circumstances.

Table 1 identifies project approaches, i.e., best practices that have been extracted from several sources, including the Software Program Managers Network 16 Point Plan, the DoD Best Practices Study conducted by Dr. Richard Turner, the European Software Institute 1977 Software Best Practice Questionnaire Analysis of Results, and various other studies. Some of the approaches are related to policy, others are related to process, and still others are related to strategy. However, they all are important considerations with significant benefits that an organization could use to establish an effective project environment and conduct the activities identified in their project plan.

As Norm Brown posits in *IEEE Software* [3], the definition of a small number of relevant best practices can have a significant effect on the success of a project, but only if the practices are tailored to the needs and culture of the organization that will use them. In Table 1, we have identified the practice; the source, including the primary reference that was used to identify it; and the general classification of the practice. Turner's dissertation [4], "Implementation of Best Practices in U.S. Department of Defense Software-Intensive Systems Acquisitions," is identified as the source for many of the practices included in the table. For readers' convenience, we have included the source reference for the specific practices identified in the Turner dissertation.

Basic Considerations for Implementing Practices

The Turner study discusses what must be

Practice Identification	Practice Source	Practice Category
Establish Clear Goals and Decision Points	(DSBTF, 5000.2R) [5,6]	Policy Requirement
Treat People as the Most Important Resource	SPMN 16 Point Plan [7]	Policy Requirement
Common Management and Manufacturing Systems	(5000.2R) [6]	Policy Requirement
Integrated Product and Process Development	(5000.2R, Reifer) [6,4]	Organizational Concept
Appointing Project Managers for Each Project	ESI [8]	Organizational Concept
Software Quality Assurance Function with Independent Reporting Line	ESI [8]	Organizational Concept
Training New Project Managers	ESI [8]	Organizational Concept
Have a Formal Review or Handover of Deliverables From One Project Group to Another	ESI [8]	Organizational Concept
Maintaining Awareness of New Development Technologies	ESI [8]	General Strategy
Ensuring User Input at All Stages of the Project	ESI [8]	General Strategy
Assess Viability, Risks, and Benefits Before Committing to a Project	ESI [8]	General Strategy
Conduct Periodic Reviews of the Status of Projects	ESI [8]	General Strategy
Demonstration-Based Reviews (Including Executable Architectures)	(Royce, DSBTF, ISO) [4,5,9]	General Strategy
Conduct Inspection and Walkthroughs at Each Stage	ESI [8]	General Strategy
Require Structured Development Methods (Iterative Processes)	(Royce, DSBTF) [4,5]	General Strategy
Plan for Technology Insertion	(5000.2R, DSMC) [6,10]	General Strategy
Commercial and Specifications and Standards/Open Systems	(5000.2R, Anderson, Jones) [6,4]	General Strategy
Statistical Process Control	Reifer [4]	General Strategy
Capture Artifacts in Rigorous, Model-Based Notation	(Royce, DSBTF) [4,5]	General Strategy
Strategic Partnering	Reifer [4]	Business Strategy
Relationship Management	Reifer [4]	Business Strategy
Market Watch	Reifer [4]	Business Strategy
Enterprise-Wide Licensing	Reifer [4]	Business Strategy
Independent Expert Reviews/SCEs	(5000.2R, DSBTF, DSMC, Jones) [6,5,10,4]	Acquisition Framework
Performance-Based Specifications	(5000.2R, Anderson, Reifer) [6, 4]	Acquisition Framework
Use Past Performance	(5000.2R, DSBTF, Anderson, Reifer) [6,5,4]	Acquisition Framework
Leverage Commercial Off-the-Shelf (COTS Items, Non-Developmental Items (NDI)	(Anderson, Reifer) [4]	Acquisition Framework
Ensure that Subcontractors Follow Formal Processes	ESI [8]	Acquisition Strategy
Best Value Awards	(5000.2R, DSBTF, Anderson, Reifer) [6,5,4]	Acquisition Strategy
Adopt Continuous Risk Management	SPMN 16 Point Plan [7]	General Practice
Estimate Empirically Cost and Schedule	SPMN 16 Point Plan [7]	General Practice
Use Metrics to Manage	SPMN 16 Point Plan [7]	General Practice
Track Earned Value	SPMN 16 Point Plan [7]	General Practice
Track Defects Against Quality Targets	SPMN 16 Point Plan [7]	General Practice
Adopt Life Cycle Configuration Management	SPMN 16 Point Plan [7]	General Practice
Manage and Trace Requirements	SPMN 16 Point Plan [7]	General Practice
Ensure Data and Database Operability	SPMN 16 Point Plan [7]	General Practice
Assess Reuse Risks and Costs	SPMN 16 Point Plan [7]	General Practice
Inspect Requirements and Design	SPMN 16 Point Plan [7]	General Practice
Requirements Trade-Off/Negotiation	(DSBTF, Royce) [5,4]	General Practice
Perform Independent Testing	ESI [8]	General Practice
Have Formal Methods of Estimating Software Size	ESI [8]	General Practice
Formally Review the Functionality of the System the Software Replaces	ESI [8]	General Practice
Use Formal Methods to Estimate Schedule and Cost	ESI [8]	General Practice
Use System-Based Software Design	SPMN 16 Point Plan [7]	Development Practice
Define and Control Interfaces	SPMN 16 Point Plan [7]	Development Practice
Design Twice, Code Once	SPMN 16 Point Plan [7]	Development Practice
Manage Testing as a Continuous Process	SPMN 16 Point Plan [7]	Development Practice
Compile and Smoke Test Frequently	SPMN 16 Point Plan [7]	Development Practice
Architecture-First Approach	(Royce, DSMC, DSBF) [4,10, 5]	Development Practice
Have Common Coding Standards for Projects	ESI [8]	Development Practice
Plan Testing Before Coding	ESI [8]	Development Practice
Establish Reliability and Stress Margins	Doherty SEI [11]	Acquisition Practice
Personal Software Process^{SM}/Team Software Process^{SM} Practices	Humphrey [12,13]	Maturity Model
Acquisition Process Improvement	SEI [14]	Maturity Model
Contractor Capability Evaluation	SEI [15]	Maturity Model

DSBTF: Defense Science Board Task Force
ESI: Enterprise Software Initiative
SCE: Software Capability Evaluation
DSMC: Defense Systems Management College

^{SM} Team Software Process and Personal Software Process are service marks of Carnegie Mellon University.

Table 1: *Examples of Best Practices*

considered to successfully migrate practices from project to project. Before management selects specific practices to accomplish a project, it must address the organization's culture, attitude, and experience, and most certainly these factors must be considered when management devises its strategy to introduce the practices selected [16].

For example, early identification and removal of defects through consistent application of structured inspections is a valuable goal. However, if the organization's management presumes that its engineers will not make mistakes, and if it will not adequately fund this practice consistently from concept through delivery, the practice is not realistic given the organization's culture.

Not only must management understand its organization's culture, it must also understand the true costs and risks associated with implementing a practice before it commits to using the practice. That is, management must honestly assess and understand the following:

- The effect of the practice on project teams regarding their possible resistance and the potential for increased productivity.
- The costs associated with the practice and the potential return on investment.
- The cost required to train those who will apply the practice.
- The availability and cost of associated tools.
- Potential barriers to implement the practice and its application.
- The validity and general acceptance of the practice within the industry.
- The effect of the practice on related and interfacing practices, processes, and tools.
- The degree of management and staff commitment to the practice, and what factors led them to commit to the practice.

It is critical that management understand the true costs and impacts of the practices it implements, whether they have been proven in other environments or not. If management implements a practice without understanding its costs and effects, it could well be incompletely or haphazardly implemented, and the project will suffer as a result. Indeed, if the implementation is incomplete, poorly planned, or otherwise improper, or, worse, if the practice must be replaced mid-project, the effects – such as poor staff morale and productivity, tool replacement, retraining, and file or artifact conversion – can be devastating.

In addition to these considerations, projects with substantial software content that show evidence of certain characteristics are poor candidates for the reasoned application of best practices. These characteristics, which were documented in the April 2002 CrossTalk [17], are the following:

- Unwarranted optimism and unrealistic executive management expectations.
- Late decision-making.
- Inappropriate use of the standard software process.
- Missing or inadequately implemented program activities.
- Lack of leadership.
- Early declarations of victory.
- An absence of risk management, which could convince managers and staff they can accomplish unrealistic objectives given the actual project circumstances.

These underlying attitudes, which can be understood to presume project success and minimal risk, often convince project management and staff that they need not adequately plan to implement a practice and develop process standards. Such complacency can be costly.

Given the fact that a best practices silver bullet does not exist, organizations cannot unthinkingly adopt a pro forma approach to project completion and assume the practices they implement will automatically succeed. To truly succeed, management must understand how the practices they use will work within their unique organization, which will lead to a solid project management foundation and will in turn positively affect the bottom line regarding productivity, quality, timeliness, and user satisfaction.◆

References

1. Turner, Ph.D., Richard. "A Study of Best Practice Adoption By Defense Acquisition Programs." CrossTalk May 2002: 4-8 <www.stsc.hill.af.mil/crosstalk/2002/05/turner.html>.
2. Brotbeck, George, Tom Miller, and Dr. Joyce Statz. "A Survey of Current Best Practices and Utilization of Standards in the Public and Private Sectors." TeraQuest Metrics, Inc., Nov., 1999.
3. Brown, Norm. "Industrial-Strength Management Strategies." IEEE Software July 1996.
4. Turner, Ph.D., Richard. "Implementation of Best Practices in U.S. Department of Defense Software-Intensive Systems Acquisitions." Washington: George Washington University, 31 Jan. 2002 <www.gold practices.com/survey/ turner/turner_dissertation.pdf>.
5. Defense Science Board. "Report of the Defense Science Board Task Force on Defense Software." Washington: Department of Defense, 2000.
6. Department of Defense. Interim Defense Acquisition Guidebook (Formerly DoD 5000.2-R). Washington: DoD, 30 Oct. 2002 <www.acq.osd.mil/dpap/Docs/InterimGuidebook.doc>.
7. Software Program Managers Network. 16 Critical Software Practices For Implementing Performance-Based Management. Vers. 3.0., Arlington, VA: Integrated Computer Engineering, Inc., 2 Aug. 2000.
8. European Software Institute. "1997 Software Best Practice Questionnaire." ESI-1997 Code Version. Bizkaia, Spain. ESI, Dec. 1997.
9. International Organization for Standardization. "Information Technology: Software Life-Cycle Processes." Geneva, Switzerland: ISO, and International Electrotechnical Commission, 1995.
10. Defense Systems Management College. "Best Practices and Course Summary." Software Acquisition Management 101. Ft. Belvoir, VA: Defense Systems Management College, 2001.
11. Doherty, Frank, and Corinne C. Segura. "Software Development Challenges Within the DoD Acquisition Framework." Systems and Software Technology Conference, Salt Lake City, UT. May 2004.
12. Humphrey, Watts. Introduction to the Personal Software Process℠. Addison-Wesley 1997.
13. Humphrey, Watts. Introduction to the Team Software Process℠. Addison-Wesley 2000.
14. Bernard, Tom, Brian Gallagher, Roger Bate, and Hal Wilson. "CMMI Acquisition Model, Vers. 1.0." CMU/SEI-2004-TR-001. Pittsburgh, PA: Software Engineering Institute, 2004 <www.sei.cmu.edu/publications/documents/04.reports/04tr001.html>.
15. Software Engineering Institute. "Word Versions of CMMI® Models." Pittsburgh, PA: SEI <www.sei.cmu.edu/cmmi/models/model-components-word.html>.
16. Evans, Michael. "Managing Acquisition Risk Through Application of Proven Best Practices." Systems and Software Technology Conference, Salt Lake City, UT. Apr., 2004.
17. Evans, Michael, Alex Abela, and Tom Beltz. "Seven Characteristics of Dysfunctional Software Projects." CrossTalk Apr. 2002 <www.stsc.hill.af.mil/crosstalk/2002/04/evans. html>.

About the Authors

Michael W. Evans is a Senior Vice President at ASC Inc. Previously he was president of Integrated Computer Engineering, Inc. He is experienced in providing direct technical services and support in software engineering methods and processes, software standards, quality assurance, configuration management, and testing. He is cofounder and prime contractor for the Software Program Managers Network, the driving force behind the Department of Defense's Software Acquisition Best Practices Initiative. He is the author of "Principles of Productive Software Management," "Productive Test Management," "Software Quality Assurance and Management," and "The Software Factory."

America Systems Corporation
88674 Shoreline Loop
Florence, OR 97439
E-mail: candca@aol.com

Corinne Segura is a Project Manager for American Systems Corporation (ASC). For the past four years, she has provided technical and managerial support in the areas of process improvement, risk management, program assessments, and test and evaluation. Prior to joining ASC, she served in the United States Navy for 20 years as a Fleet Support Officer. She has a Bachelor of Science in Biology from Northern Illinois University, a Master of Science in Systems Management from the University of Southern California and a Master of Science in Electrical Engineering from the Naval Postgraduate School.

American Systems Corporation
3033 5th AVE STE 205
San Diego, CA 92103
Phone: (619) 208-7140
E-mail: corinne.segura@2asc.com

Frank Doherty is the principal deputy for Transformation at the Space and Naval Warfare Systems Command (SPAWAR), San Diego, Calif. He has been a deputy program manager for Combat Surveillance Systems in the Program Executive Office, Command, Control, Communications, Computers, and Intelligence and Space. With 38 years of government service, Doherty has been lead for Acquisition Streamlining and chief of Industrial Quality and Productivity Division in the Office of the Secretary of Defense; chief, Contract Pricing and Financial Services Division; and director E-3A Procurement.

SPAWAR OOA-2
4301 Pacific HWY
San Diego, CA 92110-3127
Phone: (619) 524-7348
Fax: (619) 524-7374
E-mail: francis.doherty@navy.mil